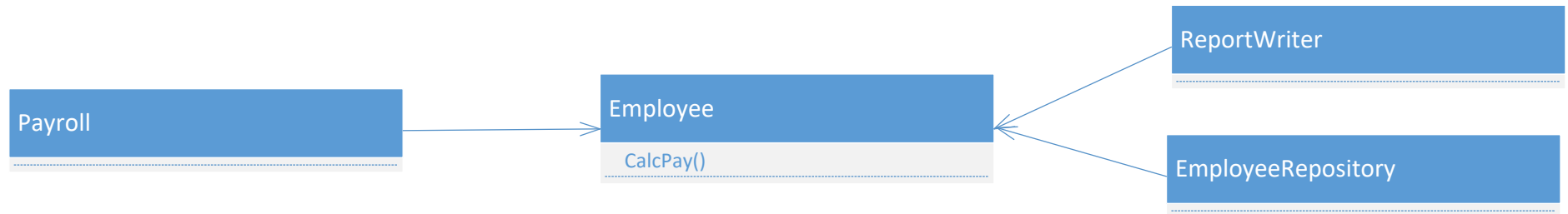
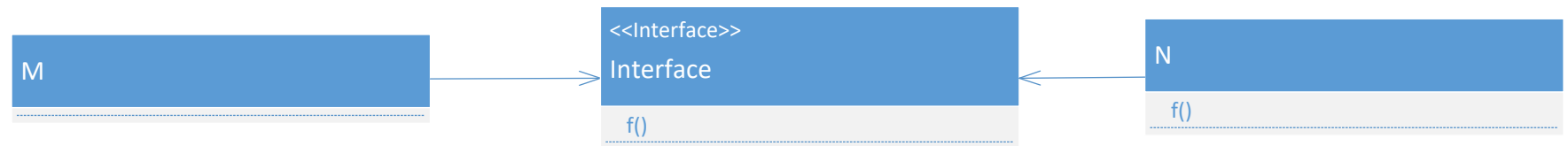


Uncle Bob's UML (Class) Diagrams



1st Shape Code (not O/C-compliant) (1:03)

Shape.h

```
enum ShapeType (circle, square);
struct Shape
{enum ShapeType itsType;};
```

Circle.h

```
struct Circle
{
    enum ShapeType itsType;
    double itsRadius;
    Point itsCenter;
};
void DrawCircle(struct Circle*)
```

Square.h

```
struct Square
{
    enum ShapeType itsType;
    double itsSide;
    Point itsTopLeft;
};
void DrawSquare(struct Square*)
```

DrawAllShapes.c

```
#include <Shape.h>
#include <Circle.h>
#include <Square.h>

typedef struct Shape* ShapePtr;

void DrawAllShapes(ShapePtr list[], int n)
{
    int i;
    for (i=0; i<n; i++ )
    {
        ShapePtr s = list[i];
        switch (s->itsType)
        {
            case square:
                DrawSquare((struct Square*)s);
                break;
            case circle:
                DrawCircle((struct Circle*)s);
                break;
        }
    }
}
```

2nd Shape Code (SOLID-OO) 1:08

Shape.h

```
Class Shape {  
public:  
    virtual void Draw() const=0;  
};
```

Square.h

```
Class Square: public Shape  
{  
public:  
    virtual void Draw() const;  
};
```

Circle.h

```
Class Circle: public Shape  
{  
public:  
    virtual void Draw() const;  
};
```

DrawAllShapes.cpp

```
#include <Shape.h>  
  
void DrawAllShapes(Shape* list[], int n)  
{  
    for(int i=0; i<n; i++)  
        list[i]->draw();  
}
```